APPENDIX D

US 7,225,282

US 7,225,282

- Title: Method and apparatus for a tow-wire serial command bus interface

US007225282B1

(12) **United States Patent**
Lyle

(10) **Patent No.: US 7,225,282 B1**
(45) **Date of Patent: May 29, 2007**

(54) **METHOD AND APPARATUS FOR A TWO-WIRE SERIAL COMMAND BUS INTERFACE**

(75) Inventor: **Jim Lyle**, Santa Clara, CA (US)

(73) Assignee: **Silicon Image, Inc.**, Sunnyvale, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1012 days.

(21) Appl. No.: **10/171,820**

(22) Filed: **Jun. 13, 2002**

(51) **Int. Cl.**
G06F 13/42 (2006.01)

(52) **U.S. Cl.** **710/105**; 710/306

(58) **Field of Classification Search** 710/100, 710/105, 306, 310; 370/401
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,818,994 A * 4/1989 Orth et al. 340/870.21
6,397,277 B1 * 5/2002 Kato et al. 710/104
6,996,112 B2 * 2/2006 Fukunaga et al. 370/401
7,060,006 B1 * 6/2006 Watterson et al. 482/54
2002/0108011 A1 * 8/2002 Tanha 710/306
2004/0049616 A1 * 3/2004 Dunstan et al. 710/100
2005/0031127 A1 * 2/2005 Gosior et al. 381/2
2005/0246475 A1 * 11/2005 Ervin 710/306

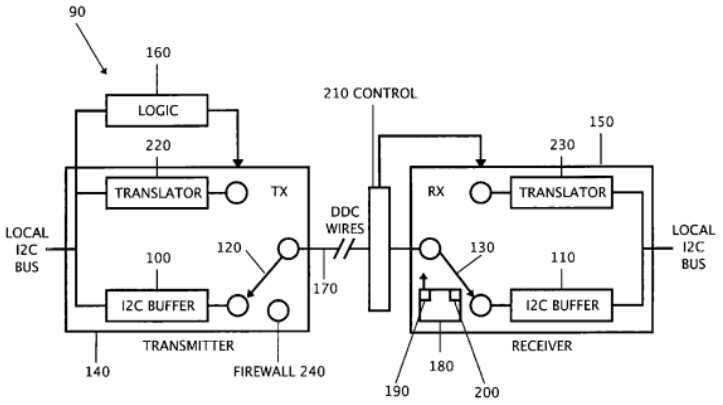
* cited by examiner

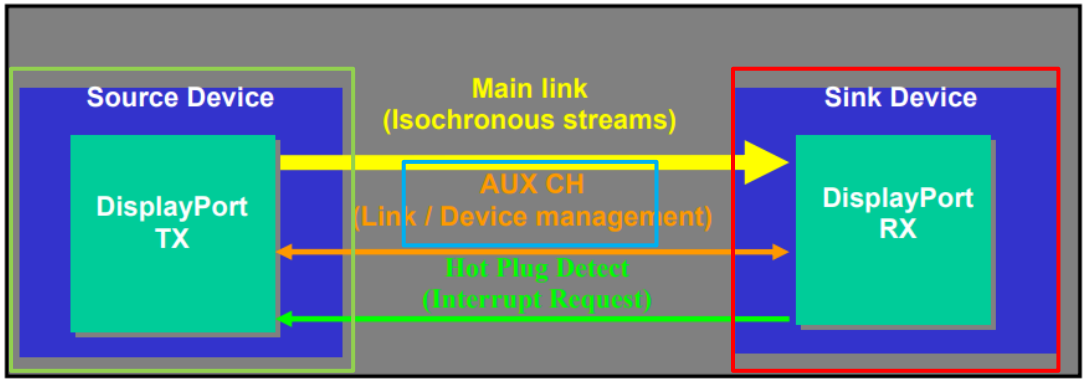
Primary Examiner—Rehana Perveen
Assistant Examiner—Clifford Knoll
(74) *Attorney, Agent, or Firm*—Perkins Coie LLP

(57) **ABSTRACT**

A method for bi-directional transmission of data between a source and a sink over a two-wire interface includes re-mapping a data signal and a clock signal from a first local bus on the source into a different protocol signal. Transmitting the different protocol signal from the source to the sink over the two-wire interface. Re-mapping the different protocol signal back into the data signal and the clock signal for use on a second local bus on the sink. Re-mapping the data signal and the clock signal from the second local bus into the different protocol signal; and transmitting the different protocol signal from the sink to the source over the two-wire interface.

37 Claims, 10 Drawing Sheets



Claim 1	VESA DisplayPort Standard v1.2
1. A method for bi-directional transmission of data between a source and a sink over a two-wire interface comprising:	**1.7 Overview of DisplayPort** A DisplayPort link consists of a main link, an auxiliary channel (AUX CH), and a Hot Plug Detect (HPD) signal line. **As shown in Figure 2-45: DisplayPort Data Transport Channels** below, the Main Link is a unidirectional, high-bandwidth and low-latency channel used below, the Main Link is a unidirectional, high-bandwidth and low-latency channel used to transport isochronous data streams such as uncompressed video and audio. The auxiliary channel is a half-duplex bidirectional channel used for link management and device control. The HPD signal also serves as an interrupt request by the Sink device. In addition, the DisplayPort connector for a box-to-box connection has a power pin for powering either a DisplayPort repeater or a DisplayPort-to-Legacy converter.  **Figure 1-1: DisplayPort Data Transport Channels**

Claim 1	VESA DisplayPort Standard v1.2
1. A method for bi-directional transmission of data between a source and a sink over a two-wire interface comprising:	***2.7 AUX Transaction Syntax in Manchester Transaction Format*** The syntaxes used for various AUX transactions in Manchester transaction format are described in this section. As described in Section 3.4, the Manchester transaction format uses Manchester-II coding at the nominal bit rate of 1Mbps. The AUX CH PHY consists of a single differential pair carrying self-clocking data. All transactions must start with a preamble "SYNC" for synchronizing the Requester (uPacket TX) and the Replier (uPacket TX), and must end with a "STOP" condition.

Claim 1	VESA DisplayPort Standard v1.2
re-mapping a data signal and a clock signal from a first local bus on the source into a different protocol signal;	***2.7 AUX Transaction Syntax in Manchester Transaction Format*** The syntaxes used for various AUX transactions in Manchester transaction format are described in this section. As described in Section 3.4, the Manchester transaction format uses Manchester-II coding at the nominal bit rate of 1Mbps. The following two categories are explained: • Native AUX transaction syntax • Syntax for mapping of an I^2C onto AUX transaction called I^2C-over-AUX transaction The AUX CH PHY consists of a single differential pair carrying self-clocking data. All transactions must start with a preamble "SYNC" for synchronizing the Requester (uPacket TX) and the Replier (uPacket TX), and must end with a "STOP" condition. A 4-bit command, COMM3:0, must be transmitted after the preamble, followed by a 20-bit address, ADDR19:0. The DisplayPort capability, status and control functions are directly mapped to the 20-bit address space. In addition, DisplayPort uses these 20 bits to access I^2C devices. After the transmission of command and address, the data bytes must be transmitted except for Address-only transaction for I^2C-over-AUX transaction. Burst data transfer is supported. The burst data size must be limited to a maximum of 16 bytes. Bit 3 (msb) of the request command field indicates whether the transaction is a Native AUX transaction or an I^2C-over-AUX transaction.

Claim 1	VESA DisplayPort Standard v1.2
re-mapping a data signal and a clock signal from a first local bus on the source into a different protocol signal;	

2.7.5 I^2C Bus Transaction Mapping onto AUX Syntax

The mapping of an I^2C transaction onto the I^2C-over-AUX transactions as defined in the DisplayPort Standard is agnostic to the application-specific usage of the I^2C data bytes. Neither the uPacket TX nor the uPacket RX must be aware of how each of the data bytes in the I^2C transaction is used for a specific I^2C application.

A single I^2C transaction may be mapped onto one or multiple I^2C-over-AUX transactions to accommodate the bit-rate difference between I^2C and AUX CH. How (or whether) to divide an I^2C transaction into multiple I^2C-over-AUX transactions is specific to the implementation of uPacket TX. For an I^2C-over-AUX

2.7.5.2 I^2C Write Transaction

In this section, mapping of an I^2C Write transaction onto the AUX transaction(s) is described using an example in which three data bytes are written. An I^2C master in the Source device will initiate an I^2C Write transaction to an I^2C slave in a Sink device via the AUX CH between uPacket TX in the Source device and uPacket RX in the Sink device as shown in Figure 2-88. Three variants of the operation are shown, demonstrating a variety of ways to accomplish the goal of performing an I^2C Write transaction.

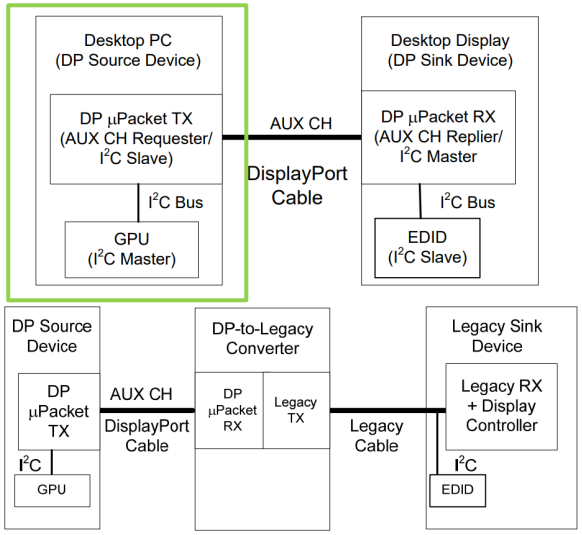


Figure 2-88: Examples of AUX CH Bridging Two I^2C Buses

Claim 1	VESA DisplayPort Standard v1.2
re-mapping a data signal and a clock signal from a first local bus on the source into a different protocol signal;	

2.7.5.2 I^2C Write Transaction

In this section, mapping of an I^2C Write transaction onto the AUX transaction(s) is described using an example in which three data bytes are written. An I^2C master in the Source device will initiate an I^2C Write transaction to an I^2C slave in a Sink device via the AUX CH between uPacket TX in the Source device and uPacket RX in the Sink device as shown in Figure 2-88. Three variants of the operation are shown, demonstrating a variety of ways to accomplish the goal of performing an I^2C Write transaction.

I^2C Write example 1:

START►1001000|0►ACK◄Data0►ACK◄Data1►ACK◄Data2►ACK◄ STOP►

Table 2-69: I^2C Write Transaction Example 1

	I^2C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I^2C Transaction in the Sink Device
1	START► 1001000|0► (I^2C Write with I^2C address = 1001000; I^2C clock stretched by uPacket TX before A			
2		...C►0100|0000► ...0000►0|1001000► STOP► (Address-only transaction, with MOT = 1 and I^2C address = 1001000)		
3			Wait up to 300µs	START► 1001000|0►ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	
5	ACK◄Data0► (I^2C clock stretched by uPacket TX before AC to Data0)			
6		...C►0100|0000► ...0000►0|1001000► 0000|0000►Data0► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
7			Wait up to 300µs	Data0►ACK◄
8			SYNC◄0000|0000◄ STOP◄ (I^2C ACK/AUX ACK)	
9	ACK◄Data1► (I^2C clock stretched by uPacket TX before ACK to Data1)			
10		...C►0100|0000► ...0000►0|1001000► ...0000►Data1► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
11			Wait up to 300µs	Data1►ACK◄
12			SYNC◄0000|0000◄ STOP◄ (I^2C ACK/AUX ACK)	
13	ACK◄Data2► (I^2C clock stretched by uPacket TX before ACK to Data2)			
14		...►0100|0000► ...00►0|1001000► ...000►Data2► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
15			Wait up to 300µs	Data2► ACK◄
16			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	
17	ACK◄STOP►			
18		...►0000|0000► ...000►0|1001000► STOP► (Address-only transaction with MOT = 0 and the same I^2C address, indicating I^2C STOP to uPacket RX)		
19			Wait up to 300µs	STOP►
20			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	

Claim 1	VESA DisplayPort Standard v1.2
re-mapping a data signal and a clock signal from a first local bus on the source into a different protocol signal;	

2.7.5.3 I^2C Read Transaction

In this section, the mapping of an I^2C read transaction onto AUX transaction(s) is described using two examples in which first example two bytes and in the second example 10 bytes are read. An I^2C master in the Source device will initiate an I^2C read transaction to an I^2C slave in the Sink device via the AUX CH between uPacket TX (in the Source device) and uPacket RX (in the Sink device).

Example 1: I^2C Read of Two Data Bytes

START►1001000|1►ACK◄Data0◄ACK►Data1◄NACK►STOP►

Table 2-72: I^2C Read Transaction Method 1

	I^2C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I^2C Transaction in Sink Device
1	START► 1001000|1► (I^2C read with I^2C addr = 1001000; I^2C Clock stretched by uPacket TX before ACK)			
2		SYNC►0101|0000► 00000000►0|1001000► STOP► (Address-only I^2C read with MOT = 1 and I^2C address = 100100)		
3			Wait up to 300µs	START► 1001000|1► ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK, I^2C address is acknowledged)	
5	ACK◄ (I^2C clock stretched by uPacket TX after ACK)			
6		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I^2C read with MOT = 1, the same I^2C address, and Length = 1 byte)		
7			Wait up to 300µs	Data0◄
8			SYNC◄0000|0000◄ Data0◄ STOP◄ (I^2C ACK / AUX ACK, sends Data0)	
9	Data0◄ACK► (I^2C clock stretched by uPacket TX after ACK)			
10		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I^2C read with MOT = 1, the same I^2C address, and Length = 1 byte)		
11				ACK►Data1◄

	I^2C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I^2C Transaction in Sink Device
12			SYNC◄0000|0000◄ Data1◄ STOP◄ (I^2C ACK / AUX ACK, sends Data1)	
13	Data1◄NACK► STOP►			
14		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (Address-only I^2C read with MOT = 0 and the same I^2C address, indicating the I2C STOP to uPacket RX)		
15			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	NACK►STOP►

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the source to the sink over the two-wire interface;	

2.7.5.2 I^2C Write Transaction

In this section, mapping of an I^2C Write transaction onto the AUX transaction(s) is described using an example in which three data bytes are written. An I^2C master in the Source device will initiate an I^2C Write transaction to an I^2C slave in a Sink device via the AUX CH between uPacket TX in the Source device and uPacket RX in the Sink device as shown in Figure 2-88. Three variants of the operation are shown, demonstrating a variety of ways to accomplish the goal of performing an I^2C Write transaction.

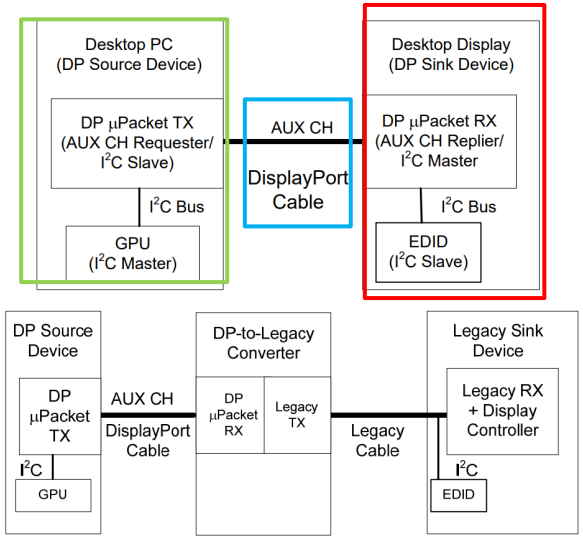


Figure 2-88: Examples of AUX CH Bridging Two I^2C Buses

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the source to the sink over the two-wire interface;	

2.7.5.2 I^2C Write Transaction

In this section, mapping of an I^2C Write transaction onto the AUX transaction(s) is described using an example in which three data bytes are written. An I^2C master in the Source device will initiate an I^2C Write transaction to an I^2C slave in a Sink device via the AUX CH between uPacket TX in the Source device and uPacket RX in the Sink device as shown in Figure 2-88. Three variants of the operation are shown, demonstrating a variety of ways to accomplish the goal of performing an I^2C Write transaction.

I^2C Write example 1:

START►1001000|0►ACK◄Data0►ACK◄Data1►ACK◄Data2►ACK◄ STOP►

	I^2C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I^2C Transaction in the Sink Device
1	START► 1001000|0► (I^2C Write with I^2C address = 1001000; I^2C clock stretched by uPacket TX before ACK)			
2		SYNC►0100|0000► 00000000►0|1001000► STOP► (Address-only transaction with MOT = 1 and I^2C address = 1001000)		
3			Wait up to 300µs	START► 1001000|0►ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	
5	ACK◄Data0► (I^2C clock stretched by uPacket TX before ACK to Data0)			
6		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data0► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
7			Wait up to 300µs	Data0►ACK◄
8			SYNC◄0000|0000◄ STOP◄ (I^2C ACK/AUX ACK)	
9	ACK◄Data1► (I^2C clock stretched by uPacket TX before ACK to Data1)			
10		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data1► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
11			Wait up to 300µs	Data1►ACK◄
12			SYNC◄0000|0000◄ STOP◄ (I^2C ACK/AUX ACK)	
13	ACK◄Data2► (I^2C clock stretched by uPacket TX before ACK to Data2)			
14		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data2► STOP► (MOT = 1, the same I^2C address, Length = 1 byte)		
15			Wait up to 300µs	Data2► ACK◄
16			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	
17	ACK◄STOP►			
18		SYNC►0000|0000► 00000000►0|1001000► STOP► (Address-only transaction with MOT = 0 and the same I^2C address, indicating I^2C STOP to uPacket RX)		
19			Wait up to 300µs	STOP►
20			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the source to the sink over the two-wire interface;	

2.7.5.3 I^2C Read Transaction

In this section, the mapping of an I^2C read transaction onto AUX transaction(s) is described using two examples in which first example two bytes and in the second example 10 bytes are read. An I^2C master in the Source device will initiate an I^2C read transaction to an I^2C slave in the Sink device via the AUX CH between uPacket TX (in the Source device) and uPacket RX (in the Sink device).

Example 1: I^2C Read of Two Data Bytes

START►1001000|1►ACK◄Data0◄ACK►Data1◄NACK►STOP►

Table 2-72: I^2C Read Transaction Method 1

	I^2C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I^2C Transaction in Sink Device
1	START► 1001000|1► (I^2C read with I^2C address = 1001000; I^2C Clock stretched by uPacket TX before ACK)			
2		SYNC►0101|0000► 00000000►0|1001000► STOP► (Address-only I^2C read with MOT = 1 and I^2C address = 100100)		
3			Wait up to 300µs	START► 1001000|1► ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK, I^2C address is acknowledged)	
5	ACK◄ (I^2C clock stretched by uPacket TX after ACK)			
6		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I^2C read with MOT = 1 and same I^2C address, and Length = 1 byte)		
7			Wait up to 300µs	Data0◄
8			SYNC◄0000|0000◄ Data0◄ STOP◄ (I^2C ACK / AUX ACK, sends Data0)	
9	Data0◄ACK► (I^2C clock stretched by uPacket TX after ACK)			
10		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I^2C read with MOT = 1, the same I^2C address, and Length = 1 byte)		
11				ACK►Data1◄
12			SYNC◄0000|0000◄ Data1◄ STOP◄ (I^2C ACK / AUX ACK, sends Data1)	
13	Data1◄NACK► STOP►			
14		SYNC►0001|0000► 00000000►0|1001000► STOP► (Address-only I^2C read with MOT = 0 and the same I^2C address, indicating the I2C STOP to uPacket RX)		
15			SYNC◄0000|0000◄ STOP◄ (I^2C ACK / AUX ACK)	NACK►STOP►

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the source to the sink over the two-wire interface;	**2.7.7 I^2C-overAUX Transaction Clarifications and Implementation Rules** This section provides clarifications to I^2C-over-AUX implementations. The objective is to eliminate interoperability issues that may be caused by varying interpretations of the specification. The syntax of the I^2C transactions is very much implementation-dependent. The syntax specification does not assume any fixed I^2C transaction syntax in an attempt to make it applicable to any I^2C implementation. The I^2C master of a uPacket RX in a Sink device will "imitate" the I2C_SCL/ I2C_DAT waveforms of those received by the I^2C slave of a uPacket TX in a Source device (with the exception of the timing of I^2C clock stretching as described in Section 2.7.7.1.6.8 and Section 2.7.7.2.6 of this document). ***2.7.7.2 Clarifications for a Sink Device*** This section describes the clarifications for a Sink device. **2.7.7.2.4 I^2C-write-over-AUX** This section describes the permissible replies by a Sink device after receiving an I^2C-write-over-AUX request transaction from a Source device. A Sink device must reply with one of the ways as described in this section. **2.7.7.2.5 I^2C -read-over-AUX** This section describes the permissible replies by a Sink device after receiving an I^2C-read-over-AUX request transaction from a Source device. A Sink device must reply with one of the ways as shown in this section.

Claim 1	VESA DisplayPort Standard v1.2
re-mapping the different protocol signal back into the data signal and the clock signal for use on a second local bus on the sink;	**2.7.6 Conversion of I²C Transaction to Native AUX Transaction (Informative)** Conversion of an I²C transaction into a Native AUX transaction by the uPacket TX is implementation-specific and is beyond the scope of this Standard. When the mapping of I²C transaction over the AUX CH, the translation of I²C to AUX transaction by the uPacket TX and that of the AUX to the I²C by the uPacket RX must agree with each other. Therefore, the translation mechanism is defined in this Standard. **2.7.7 I²C-overAUX Transaction Clarifications and Implementation Rules** This section provides clarifications to I²C-over-AUX implementations. The objective is to eliminate interoperability issues that may be caused by varying interpretations of the specification. The syntax of the I²C transactions is very much implementation-dependent. The syntax specification does not assume any fixed I²C transaction syntax in an attempt to make it applicable to any I²C implementation. The I²C master of a uPacket RX in a Sink device will "imitate" the I2C_SCL/ I2C_DAT waveforms of those received by the I²C slave of a uPacket TX in a Source device (with the exception of the timing of I²C clock stretching as described in Section 2.7.7.1.6.8 and Section 2.7.7.2.6 of this document).

Claim 1	VESA DisplayPort Standard v1.2
re-mapping the data signal and the clock signal from the second local bus into the different protocol signal; and	

2.7.5 I²C Bus Transaction Mapping onto AUX Syntax

The mapping of an I²C transaction onto the I²C-over-AUX transactions as defined in the DisplayPort Standard is agnostic to the application-specific usage of the I²C data bytes. Neither the uPacket TX nor the uPacket RX must be aware of how each of the data bytes in the I²C transaction is used for a specific I²C application.

A single I²C transaction may be mapped onto one or multiple I²C-over-AUX transactions to accommodate the bit-rate difference between I²C and AUX CH. How (or whether) to divide an I²C transaction into multiple I²C-over-AUX transactions is specific to the implementation of uPacket TX. For an I²C-over-AUX

Table 2-69: I²C Write Transaction Example 1

	I²C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in the Sink Device
1	START► 1001000|0► (I²C Write with I²C address = 1001000; I²C clock stretched by uPacket TX before ACK)			
2		SYNC►0100|0000► 00000000►0|1001000► STOP► (Address-only transaction, with MOT = 1 and I²C address = 1001000)		
3			Wait up to 300µs	START 100 ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	
5	ACK◄Data0► (I²C clock stretched by uPacket TX before ACK to Data0)			
6		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data0► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
7			Wait up to 300µs	CK◄
8			SYNC◄0000|0000◄ STOP◄ (I²C ACK/AUX ACK)	
9	ACK◄Data1► (I²C clock stretched by uPacket TX before ACK to Data1)			
10		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data1► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
11			Wait up to 300µs	Data1►ACK◄

	I²C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in the Sink Device
12			SYNC◄0000|0000◄ STOP◄ (I²C ACK/AUX ACK)	
13	ACK◄Data2► (I²C clock stretched by uPacket TX before ACK to Data2)			
14		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data2► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
15			Wait up to 300µs	ACK◄
16			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	
17	ACK◄STOP►			
18		SYNC►0000|0000► 00000000►0|1001000► STOP► (Address-only transaction with MOT = 0 and the same I²C address, indicating I²C STOP to uPacket RX)		
19			Wait up to 300µs	STOP►
20			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	

Claim 1	VESA DisplayPort Standard v1.2
re-mapping the data signal and the clock signal from the second local bus into the different protocol signal; and	

2.7.5 I²C Bus Transaction Mapping onto AUX Syntax

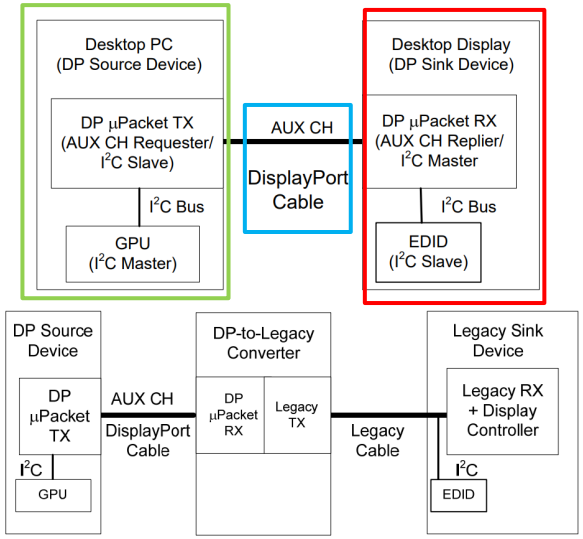
The mapping of an I²C transaction onto the I²C-over-AUX transactions as defined in the DisplayPort Standard is agnostic to the application-specific usage of the I²C data bytes. Neither the uPacket TX nor the uPacket RX must be aware of how each of the data bytes in the I²C transaction is used for a specific I²C application.

A single I²C transaction may be mapped onto one or multiple I²C-over-AUX transactions to accommodate the bit-rate difference between I²C and AUX CH. How (or whether) to divide an I²C transaction into multiple I²C-over-AUX transactions is specific to the implementation of uPacket TX. For an I²C-over-AUX

Table 2-72: I²C Read Transaction Method 1

	I²C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in Sink Device
1	START▶ 1001000|1▶ (I²C read with I²C address = 1001000; I²C Clock stretched by uPacket TX before ACK)			
2		SYNC▶0101|0000▶ 00000000▶0|1001000▶ STOP▶ (Address-only I²C read with MOT = 1 and I²C address = 100100)		
3			Wait up to 300µs	START▶ 1001000|1▶ ACK◀
4			SYNC◀0000|0000◀ STOP◀ (I²C ACK / AUX ACK, I²C address is acknowledged)	
5	ACK◀ (I²C clock stretched by uPacket TX after ACK)			
6		SYNC▶0101|0000▶ 00000000▶0|1001000▶ 0000|0000▶STOP▶ (I²C read with MOT = 1, the same I²C address, and Length = 1 byte)		
7			Wait up to 300µs	Data0◀
8			SYNC◀0000|0000◀ Data0◀ STOP◀ (I²C ACK / AUX ACK, sends Data0)	
9	Data0◀ACK▶ (I²C clock stretched by uPacket TX after ACK)			
10		SYNC▶0101|0000▶ 00000000▶0|1001000▶ 0000|0000▶STOP▶ (I²C read with MOT = 1, the same I²C address, and Length = 1 byte)		
11				ACK▶Data1◀

	I²C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in Sink Device
12			SYNC◀0000|0000◀ Data1◀ STOP◀ (I²C ACK / AUX ACK, sends Data1)	
13	Data1◀NACK▶ STOP▶			
14		SYNC▶0001|0000▶ 00000000▶0|1001000▶ STOP▶ (Address-only I²C read with MOT = 0 and the same I²C address, indicating the I2C STOP to uPacket RX)		
15			SYNC◀0000|0000◀ STOP◀ (I²C ACK / AUX ACK)	NACK▶STOP▶

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the sink to the source over the two-wire interface.	**2.7.2 AUX Transaction Response/Reply Time-outs** AUX Replier (the uPacket RX) must start sending the reply back to the AUX requester (the uPacket TX) within the response period of 300µs. The timer for Response Time-out starts ticking after the uPacket RX has finished receiving the AUX STOP condition which ends the AUX Request transaction. The timer is reset either when the Response Time-out period has elapsed or when the uPacket RX has started to send the AUX Sync pattern (which follows 10 to 16 active pre-charge pulses) for the Reply transaction. If the uPacket TX does not receive a reply from the uPacket RX it must wait for a Reply Time-out period of 400us before initiating the next AUX Request transaction. The timer for the Reply Time-out starts ticking after the uPacket TX has finished sending the AUX STOP condition. The timer is reset either when the Reply Time-out period elapses or when the uPacket TX detects the first zero in Manchester-II code which is in active pre-charge pulses and AUX Sync pattern of the Reply transaction.  **Figure 2-88: Examples of AUX CH Bridging Two I²C Buses**

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the sink to the source over the two-wire interface.	

2.7.5 I²C Bus Transaction Mapping onto AUX Syntax

The mapping of an I²C transaction onto the I²C-over-AUX transactions as defined in the DisplayPort Standard is agnostic to the application-specific usage of the I²C data bytes. Neither the uPacket TX nor the uPacket RX must be aware of how each of the data bytes in the I²C transaction is used for a specific I²C application.

A single I²C transaction may be mapped onto one or multiple I²C-over-AUX transactions to accommodate the bit-rate difference between I²C and AUX CH. How (or whether) to divide an I²C transaction into multiple I²C-over-AUX transactions is specific to the implementation of uPacket TX. For an I²C-over-AUX

	I²C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in the Sink Device
1	START► 1001000|0► (I²C Write with I²C address = 1001000; I²C clock stretched by uPacket TX before ACK)			
2		SYNC►0100|0000► 00000000►0|1001000► STOP► (Address-only transaction, with MOT = 1 and I²C address = 1001000)		
3			Wait up to 300µs	START► 1001000|0►ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	
5	ACK◄Data0► (I²C clock stretched by uPacket TX before ACK to Data0)			
6		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data0► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
7			Wait up to 300µs	Data0►ACK◄
8			SYNC◄0000|0000◄ ◄ ACK/AUX ACK)	
9	ACK◄Data1► (I²C clock stretched by uPacket TX before ACK to Data1)			
10		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data1► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
11			Wait up to 300µs	Data1►ACK◄

	I²C Transaction in the Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in the Sink Device
12			SYNC◄0000|0000◄ OP◄ C ACK/AUX ACK)	
13	ACK◄Data2► (I²C clock stretched by uPacket TX before ACK to Data2)			
14		SYNC►0100|0000► 00000000►0|1001000► 0000|0000►Data2► STOP► (MOT = 1, the same I²C address, Length = 1 byte)		
15			Wait up to 300µs	Data2► ACK◄
16			C◄0000|0000◄ P◄ (I²C ACK / AUX ACK)	
17	ACK◄STOP►			
18		SYNC►0000|0000► 00000000►0|1001000► STOP► (Address-only transaction with MOT = 0 and the same I²C address, indicating I²C STOP to uPacket RX)		
19			Wait up to 300µs	STOP►
20			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	

Claim 1	VESA DisplayPort Standard v1.2
transmitting the different protocol signal from the sink to the source over the two-wire interface.	

2.7.5 I²C Bus Transaction Mapping onto AUX Syntax

The mapping of an I²C transaction onto the I²C-over-AUX transactions as defined in the DisplayPort Standard is agnostic to the application-specific usage of the I²C data bytes. Neither the uPacket TX nor the uPacket RX must be aware of how each of the data bytes in the I²C transaction is used for a specific I²C application.

A single I²C transaction may be mapped onto one or multiple I²C-over-AUX transactions to accommodate the bit-rate difference between I²C and AUX CH. How (or whether) to divide an I²C transaction into multiple I²C-over-AUX transactions is specific to the implementation of uPacket TX. For an I²C-over-AUX

Table 2-72: I²C Read Transaction Method 1

	I²C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in Sink Device
1	START► 1001000|1► (I²C read with I²C address = 1001000; I²C Clock stretched by uPacket TX before ACK)			
2		SYNC►0101|0000► 00000000►0|1001000► STOP► (Address-only I²C read with MOT = 1 and I²C address = 100100)		
3			Wait up to 300µs	START► 1001000|1► ACK◄
4			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK, I²C address is acknowledged)	
5	ACK◄ (I²C clock stretched by uPacket TX after ACK)			
6		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I²C read with MOT = 1, the same I²C address, and Length = 1 byte)		
7			Wait up to 300µs	Data0◄
8			SYNC◄0000|0000◄ Data0◄ STOP◄ (I²C ACK / AUX ACK, sends Data0)	
9	Data0◄ACK► (I²C clock stretched by uPacket TX after ACK)			
10		SYNC►0101|0000► 00000000►0|1001000► 0000|0000►STOP► (I²C read with MOT = 1, the same I²C address, and Length = 1 byte)		
11				ACK►Data1◄

	I²C Transaction in Source Device	AUX Request Transaction by uPacket TX	AUX Reply Transaction by uPacket RX	I²C Transaction in Sink Device
12			SYNC◄0000|0000◄ Data1◄ STOP◄ (I²C ACK / AUX ACK, sends Data1)	
13	Data1◄NACK► STOP►			
14		SYNC►0001|0000► 00000000►0|1001000► STOP► (Address-only I²C read with MOT = 0 and the same I²C address, indicating the I2C STOP to uPacket RX)		
15			SYNC◄0000|0000◄ STOP◄ (I²C ACK / AUX ACK)	NACK►STOP►